



Sztuczna inteligencja i inżynieria wiedzy

Laboratorium - Lista 3

6.06.2024

Krzysztof Głowacz, 266545

Spis treści

1	Wstęp	2
2	Sieć semantyczna	3
3	Model faktów w bazie wiedzy	3
4	Przykład wnioskowania	4
5	Wypracowanie rozwiązania	4
6	Możliwości bazy wiedzy	5
7	Wnioski	6

1 Wstęp

Laboratorium nr 3 dotyczyło reprezentacji wiedzy oraz uruchamiania wnioskowania. Wiedza opisana została w postaci faktów i reguł, a następnie uruchamiany był proces wnioskowania w przód (*Forward Chaining*) bazujący na algorytmie *Rete*. Do realizacji zadania wykorzystano bibliotekę *Experta*, która pozwala za pomocą predefiniowanych wzorców i makr budować proste bazy wiedzy i przeprowadzać wnioskowanie.

Urządzeniem wybranym do zaprezentowania możliwości biblioteki *Experta* został ekspres do kawy *Nivona Cafe Romantica 550/560*, a konkretniej sekcja dot. samodzielnego rozwiązywania problemów na stronie 25. w instrukcji obsługi [1]. Opisana ona została w postaci przejrzystej tabeli, którą przedstawiają poniższe zdjęcia (rozbita została na dwa osobne zrzuty ekranu ze względu na swój rozmiar).

Problem	Przyczyna	Usunięcie problemu
Wytwarzanie gorącej wody lub spienionego mleka / pary niemożliwe	Możliwe zakamienienie systemu	Przeprowadzić odkamienianie z wysoką dawką środka (rozdział 8.5.)
	Spumatore jest zatkany	Dokładnie oczyścić spumatore, całkowicie rozkładając go na części
Zbyt mało spienionego mleka lub mleko słabo spienione	Spumatore jest zatkany	Dokładnie oczyścić spumatore, całkowicie rozkładając go na części
	Nieodpowiednie mleko	Używać zimnego mleka
	Możliwe zakamienienie systemu	Przeprowadzić odkamienianie z wysoką dawką środka (rozdział 8.5.)
Kawa wypływa jedynie kroplami	Możliwe zakamienienie systemu	Przeprowadzić odkamienianie z wysoką dawką środka (rozdział 8.5.)
	Zbyt drobny / zbyt gruby stopień zmielenia	Ustawić grubszy/drobniejszy stopień zmielenia (rozdział 7.2)
	Zbyt drobno zmielona kawa	Użyć grubiej zmielonej kawy
	Wsypało za dużo kawy mielonej	Wsypać mniej kawy mielonej
	Zaparzacz zatkany	Wyjąć i oczyścić zaparzacz (rozdział 8.6.). Ponownie włożyć zaparzacz i przeprowadzić czyszczenie systemu (rozdział 8.4.)
	Wyloty kawy są zatkane	Przeprowadzić czyszczenie systemu (rozdział 8.4.)

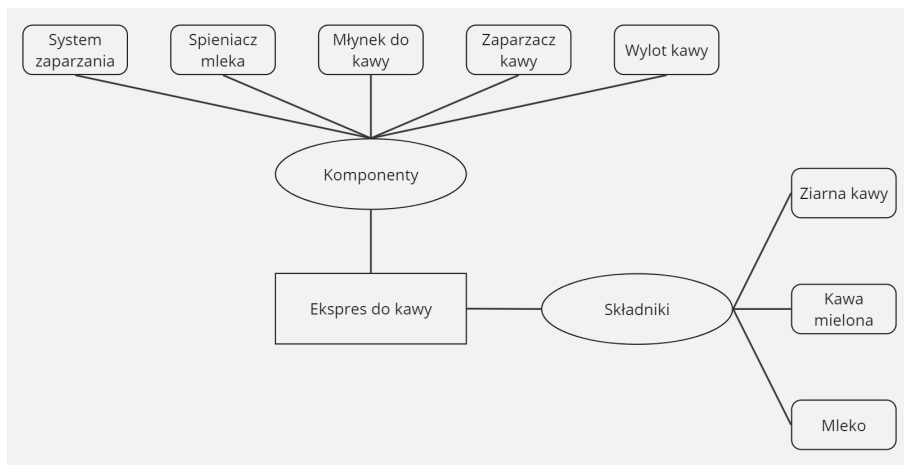
Zdjęcie 1: Instrukcja rozwiązywania problemów - cz. 1

Problem	Przyczyna	Usunięcie problemu
Brak „pianki” na kawie	Zaparzacz zatkany	Wyjąć i oczyścić zaparzacz (rozdział 8.6.). Ponownie włożyć zaparzacz i przeprowadzić czyszczenie systemu (rozdział 8.4.).
	Nieodpowiedni rodzaj ziaren kawy	Zmienić rodzaj ziaren kawy
	Ziarna nie są świeżo palone	Użyć świeżych ziaren
Głośna praca młynka	Stopień zmielenia nie jest dopasowany do ziaren kawy	Zoptymalizować stopień zmielenia (rozdział 7.2.)
	Ciała obce, np. kamyki, w młynku	Zadzwonić na infolinię serwisową NIVONA; nadal można zaparzać kawę mieloną
Na wyświetlaczu pojawia się Błąd 8 tel. serwisowy	Nie można prawidłowo pozycjonować zaparzacza przed napędem	Wyłączyć urządzenie i wyciągnąć wtyczkę. Ponownie włożyć wtyczkę i włączyć urządzenie. Gdy urządzenie będzie gotowe do pracy, wyjąć i oczyścić zaparzacz (rozdział 8.6.).

Zdjęcie 2: Instrukcja rozwiązywania problemów - cz. 2

2 Sieć semantyczna

Po zapoznaniu się z tabelą ze zdjęć 1 i 2 można zdefiniować prostą sieć semantyczną, która przedstawia pojęcia związane z procesem rozwiązywania problemów występujących w tym konkretnym modelu ekspresu do kawy. Propozycja takiej sieci wygląda następująco:



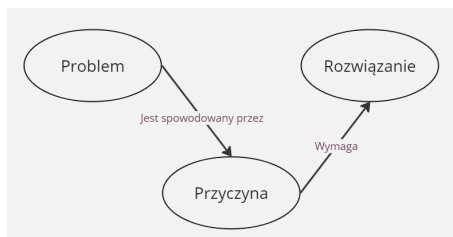
Zdjęcie 3: Sieć semantyczna ekspresu do kawy

3 Model faktów w bazie wiedzy

Kolejnym etapem podczas niniejszego laboratorium było zamodelowanie faktów w bazie wiedzy, na podstawie których uruchamiany będzie silnik służący do przeprowadzania samego wnioskowania. Instrukcja rozwiązywania problemów przygotowania jest przez producenta w taki sposób, że aby znaleźć rozwiązanie problemu należy wskazać zauważony problem, a następnie odpowiedzieć na kilka pytań „Tak”/„Nie”, dzięki którym wykrywana jest przyczyna i sugerowane jest rozwiązanie. Aby efektywnie przeprowadzać wnioskowanie zdefiniowane zostały więc fakty:

- *Problem*
- *Cause*
- *Solution*

Dodatkowo *Cause* ma pole przechowujące stan sprawdzenia danej przyczyny. Domyślnie wszystkie przyczyny mają przypisany stan **IRRELEVANT**, tzn. traktowane są jako nieistotne w danej sytuacji. Podczas działania algorytmu stan ten może zostać zmieniony na jeden z trzech pozostałych: **INVESTIGATING**, **CONFIRMED** lub **DENIED**. Taki sposób przedstawienia problemu pozwala nam ogólnie podejść do procesu szukania rozwiązania - przy ogólnie określonym problemie poszukiwane będą możliwe przyczyny danego błędu, a na podstawie zmian ich stanu podejmowane będą kolejne kroki (albo algorytm uzna, że przyczyna została znaleziona, więc przypisana jej akcja powinna być poszukiwanym rozwiązaniem, albo po odrzuceniu danej przyczyny poszukiwania będą kontynuowane). Symboliczny schemat takiego modelu wygląda następująco:



Zdjęcie 4: Symboliczny diagram modelu faktów bazy wiedzy

4 Przykład wnioskowania

Przykładowe reguły wnioskowania dla problemu **Brak „pianki” na kawie** zgodne z opisanym wyżej modelem:

1. Zmień stan przyczyny *Zaparzacz zatkany* z **IRRELEVANT** na **INVESTIGATING**.
2. Jeśli zaparzacz jest zatkany - zmień stan przyczyny na **CONFIRMED** i zakończ wnioskowanie wyświetleniem znalezionego rozwiązania.
3. Jeśli zaparzacz nie jest zatkany - zmień stan przyczyny na **DENIED** oraz stan przyczyny *Nieodpowiedni rodzaj ziaren kawy* z **IRRELEVANT** na **INVESTIGATING**.
4. Jeśli rodzaj ziaren kawy jest nieodpowiedni - zmień stan przyczyny na **CONFIRMED** i zakończ wnioskowanie wyświetleniem znalezionego rozwiązania.
5. Jeśli rodzaj ziaren kawy jest odpowiedni - zmień stan przyczyny na **DENIED** oraz stan przyczyny *Ziarna nie są świeżo palone* z **IRRELEVANT** na **INVESTIGATING**.
6. Jeśli ziarna kawy nie są świeżo palone - zmień stan przyczyny na **CONFIRMED** i zakończ wnioskowanie wyświetleniem znalezionego rozwiązania.
7. Jeśli ziarna kawy są świeżo palone - zmień stan przyczyny na **DENIED**.
8. Jeśli wszystkie potencjalne przyczyny mają stan **IRRELEVANT** lub **DENIED** - zakończ wnioskowanie komunikatem o konieczności kontaktu z autoryzowanym serwisem producenta.

5 Wypracowanie rozwiązania

Podczas rozwiązywania niniejszej listy zadań zmieniała się koncepcja faktów i reguł - początkowo faktami miały być konkretne komponenty ekspresu, Szybko się jednak okazało, że wystarczy opisać potencjalne przyczyny problemów, ponieważ one są częścią tabeli przygotowanej przez producenta, niekiedy się powtarzają i pozwalają minimalnym nakładem kodu zdefiniować jednocześnie pytania, na które użytkownik programu musi odpowiedzieć. Następnie w wyniku wprowadzania kolejnych usprawnień dodane zostało pole przechowujące stan sprawdzenia danej przyczyny - pozwala ono ponownie zredukować wiele kodu - wystarczy bowiem zdefiniować jedną funkcję odpowiedzialną za automatyczną zmianę stanu na podstawie odpowiedzi użytkownika na **CONFIRMED** lub **DENIED**. Inna funkcja z kolei, korzystająca z instrukcji *FORALL* wbudowanej w bibliotekę *Experta* pozwala na automatyczne sprawdzanie, czy nie wystąpiła sytuacja, w której wszystkie przyczyny mają przypisany stan **IRRELEVANT** lub **DENIED** - wówczas należy przerwać wnioskowanie. Jeszcze inna funkcja sprawdza, czy istnieje jakakolwiek przyczyna, która ma przypisany stan **CONFIRMED** - to także jest sygnał dla programu, aby zakończyć wnioskowanie. Dla wszystkich rozpatrywanych problemów nie trzeba więc osobno sprawdzać, czy znaleziona została konkretna przyczyna lub czy nie udało się znaleźć żadnego rozwiązania. Robią to poniższe funkcje:

```
@Rule(Cause(description=MATCH.description, status=CauseStatus.CONFIRMED))
def diagnose_solution(self, description):
    self.declare(Solution(action=symptom_to_action_map[description]))

@Rule(
    FORALL(
        Cause(is_present=MATCH.status),
        TEST(lambda status: status in [CauseStatus.IRRELEVANT.value, CauseStatus.DENIED.value])
    )
)
def diagnose_unknown_issue(self):
    self.declare(Solution(action=Actions.CONTACT_SERVICE.value))
```

Jedynie reguły jakie należy definiować dla każdego z problemów to te, które określają kolejne potencjalne przyczyny problemu i przejścia między nimi w momencie, gdy nie udało się dotychczas znaleźć rozwiązania.

6 Możliwości bazy wiedzy

Na koniec laboratorium należało sprawdzić możliwości stworzonej bazy wiedzy i reguł wnioskowania. Dzięki opisanej powyżej metodzie definicji poszczególnych elementów biorących udział w procesie rozwiązywania problemów z ekspresem do kawy możliwe jest kontekstowe zadawanie pytań użytkownikowi, które mają na celu zawężenie obszaru poszukiwań problemu. Z każdą kolejną odpowiedzią negatywną zmniejsza się liczba pozostałych potencjalnych rozwiązań danego błędu systemu. Po dojściu do końca, tzn. w sytuacji, gdy rozwiązanie nie zostało znalezione, a lista możliwych przyczyn problemu się wyczerpała - wyświetlany jest jeden komunikat wspólny dla wszystkich problemów informujący, że należy skontaktować się z autoryzowanym serwisem. Jeśli po drodze padnie odpowiedź twierdząca, tzn. użytkownik potwierdzi, że dana sytuacja ma miejsce, wtedy program zwraca znaną przyczynę i sugeruje rozwiązanie problemu. Przykładowe uruchomienia programu prezentują poniższe zrzuty ekranu:

```
o Choose the problem:
o [1] Hot water or steam generation impossible
  [2] Milk froth is too weak or no milk froth
  [3] Coffee flows only drop by drop
  [4] No crema on the coffee
  [5] Loud grinder noise
  [6] Error message on display 'Error 8 service call'

Enter the number of the problem: 1

Is system scaled? (y/n): n
Is spumatore clogged? (y/n): y

Detected cause: spumatore clogged

Solution: Thoroughly clean the spumatore, disassemble it completely
```

Zdjęcie 5: Przykład 1. - *Wytwarzanie gorącej wody lub spienionego mleka/pary niemożliwe*

```
o Choose the problem:
  [1] Hot water or steam generation impossible
  [2] Milk froth is too weak or no milk froth
  [3] Coffee flows only drop by drop
  [4] No crema on the coffee
  [5] Loud grinder noise
  [6] Error message on display 'Error 8 service call'

Enter the number of the problem: 3

Is system scaled? (y/n): n
Is grind size inappropriate (too fine or too coarse)? (y/n): n
Is ground coffee very fine [if used]? (y/n): n
Is added too much ground coffee [if used]? (y/n): y

Detected cause: added too much ground coffee [if used]

Solution: Use less ground coffee

Are you satisfied with the provided solution and want to exit the troubleshooting assistant? (y/n):
```

Zdjęcie 6: Przykład 2. - *Kawa wypływa jedynie kroplami*

```
o Choose the problem:
  [1] Hot water or steam generation impossible
  [2] Milk froth is too weak or no milk froth
  [3] Coffee flows only drop by drop
  [4] No crema on the coffee
  [5] Loud grinder noise
  [6] Error message on display 'Error 8 service call'

Enter the number of the problem: 4

Is brewing unit clogged? (y/n): n
Is type of coffee beans inappropriate? (y/n): n
Are coffee beans NOT freshly roasted? (y/n): n

Solution: Seems like a bigger issue - please contact professional service
```

Zdjęcie 7: Przykład 3. - *Brak „pianki” na kawie*

```

Choose the problem:
[1] Hot water or steam generation impossible
[2] Milk froth is too weak or no milk froth
[3] Coffee flows only drop by drop
[4] No crema on the coffee
[5] Loud grinder noise
[6] Error message on display 'Error 8 service call'

Enter the number of the problem: 5

Is grind size NOT adjusted properly to the coffee beans (too fine / too coarse)? (y/n): n
Are foreign objects in grinder? (y/n): y

Detected cause: foreign objects in grinder

Solution: Remove the foreign objects or contact service

```

Zdjęcie 8: Przykład 4. - *Głośna praca młynka*

7 Wnioski

Tak przygotowana baza wiedzy z faktami i regułami stanowi kompletną podstawę do ewentualnej dalszej rozbudowy modelu rozwiązywania problemów. Ogromną zaletą takiego podejścia, która stanowi jego przewagę nad standardowym pisanem warunków w blokach *if - else* jest fakt, że każda reguła jest tutaj osobną częścią, naturalnie wydzieloną logiką. To pozwala na łatwe modyfikowanie reguł, ich dodawanie czy usuwanie bez jakiegokolwiek modyfikacji pozostałej części kodu. Przy większych systemach zapisanie wszystkich powiązań za pomocą bloków *if - else* byłoby niezwykle problematyczne ze względu na nieczytelność takiego kodu i trudność w jego utrzymaniu i modyfikacji. Bazę wiedzy tego typu można natomiast wygodnie skalować w dowolnym momencie. Nie ma konieczności także określać przepływu danych i przechodzenia po drzewie decyzyjnym - za to odpowiada silnik, któremu do działania wystarczą deklaratorywnie tworzone reguły.

Literatura

- [1] Instrukcja obsługi ekspresu do kawy
https://nivona.pl/media/93/e4/d5/1652945780/BDA%20NICR%205.%.%20Rev.01%2001.04.2022_eBook_PL.pdf. Dostęp: 1.06.2024.